

AUTOMATED ADMINISTRATION SYSTEM
FOR STATE-BASED CONTROL OF A TERMINAL USER INTERFACE

Related Applications

~~The present application is related to U.S. Patent Application entitled "Feature Access Control in a Display-Based Terminal Environment," and U.S. Patent Application entitled "State-Based Control of a Terminal User Interface Containing Soft-Labeled Keys," both filed concurrently herewith.~~

Field of the Invention

The invention relates generally to communication systems, and more particularly to communication systems which support wireless mobile telephones or other types of user terminals which include soft-labeled keys (SLKs).

Background of the Invention

Conventional communication systems may be configured to support wireless terminals which utilize SLKs. The functions associated with such keys can be varied under control of a system switch or switch adjunct, such that the same physical key can represent multiple features at different times. This compensates for the typical lack of user interface "real estate" on the wireless terminal by providing full feature access even with many fewer physical keys than, e.g., a corresponding wired terminal supported by the same system. A wireless terminal which incorporates SLKs generally includes a display containing the labels associated with the SLKs. In a conventional premises switching system, updates of these labels are typically explicitly controlled by the switch, e.g., based on predetermined functional modes associated with an operating context of the wireless terminal, and/or in response to commands entered by a user at the wireless terminal.

A significant problem with providing such a context-sensitive soft-labeled wireless terminal is that a number of run time misoperations or service degradations can occur if conventional command and update strategies are used to drive the wireless terminal. For example, if the system switch provides updates on a per-key-depression basis, the switch expends a considerable portion of its processing capacity in simply updating the label context of the SLKs on the wireless terminal.

This is undesirable since it reduces system capacity, and since it introduces potentially unacceptable delays in updating the key labels. The latter difficulty may also lead to an interpretative misoperation. For example, assume the user depresses multiple SLKs at the wireless terminal. As a result, the switch sends a collection of updates to the terminal, and the first SLK update is processed and displayed. However, if the user then depresses another one of the SLKs, the switch has no way of knowing if all of the updates have been processed at the terminal, and therefore must impose interpretive assumptions about the terminal labels being displayed at the point in time when an SLK is depressed. This is an undesirable interpretive race condition, since the switch is mapping terminal button identifiers to system feature codes and the identifiers and codes may be desynchronized. Another problem associated with conventional control of SLKs is that a significant amount of bandwidth can be expended in the process of transmitting updates to the wireless terminal, thereby reducing the local radio access efficiency of the system.

Additional problems arise in conventional systems with regard to feature access control. The SLK display line in a terminal of such a system will generally present those features that are of the most use to the user in any given context. This in turn requires that the same area of the display present different features at different times, and in different contexts. For example, a terminal in an IDLE condition would have no use for a call management feature such as HOLD, while a feature such as a directory service would be of value. Therefore, the display must provide different SLK labels during different conditions. Also note that, depending on the service mode, the same system feature may be required in numerous contexts. For example, the HOLD feature may be of value during any incoming or outgoing call, and may also be of value during conferencing and transferring operations. This being the case, the HOLD feature will have four occurrences in a dynamically changing feature mix presented to the user via the SLKs. Conventional techniques are generally unable to provide efficient control of feature access. Such techniques may require an excessive number of multiple update transmissions from the serving switch to the terminal, thereby increasing the bandwidth required to support the terminal user interface.

A need therefore exists for techniques which allow a communication system switch to control SLKs of a wireless terminal user interface in a more efficient manner, while avoiding the run time misoperations and other problems associated with conventional techniques.

Summary of the Invention

The invention provides an automated administration system for providing state-based control of soft-labeled keys (SLKs) in a wireless terminal or other type of communication system terminal. In an illustrative embodiment, the automated administration system uses a set of operations to generate information representative of a state machine for controlling labels for the SLKs. The operations process input received from a given user, e.g., a form specifying desired features, layout and language, and generate a state transition table or other suitable representation of a state machine which provides those features, layout and language. Other information generated by the automated administration system may include a control table and a label table. The control table associates a different set of SLK label identifiers with each state in a set of states of the terminal. Each of the label identifiers specifies a label to be associated with a given one of the SLKs in one or more of the states. The label identifiers are used as pointers into the label table which specifies, for each of the label identifiers, a corresponding label for one of the SLKs. The set of operations may be repeated for different users or groups of users of the system, such that a different state machine is generated for each of the users or groups, thereby allowing different users or groups to have different types and arrangements of feature access via their terminal SLKs. The automated administration system and its set of operations may be implemented at least in part in software associated with a switch of the system.

The automated administration aspects of the invention provide a number of advantages over conventional systems. For example, the invention in the illustrative embodiment performs state transition table creation on a per-user and/or per-group basis, and loads into the table all operating data for the specific feature layout associated with a given user or group. The invention significantly reduces the costs associated with system administration, while improving system performance and flexibility. These and other features and advantages of the present invention will become more apparent from the accompanying drawings and the following detailed description.

Brief Description of the Drawings

FIG. 1 shows a portion of an exemplary communication system in which the invention may be implemented.

FIG. 2 shows an illustrative embodiment of a wireless terminal configured in accordance with the invention.

FIGS. 3 and 4 are diagrams illustrating the operating modes and state transitions, respectively, in the wireless terminal of FIG. 2 in accordance with the invention.

5 FIG. 5 shows a portion of a state transition table in accordance with the invention.

FIGS. 6A and 6B show a more detailed example of a state transition table in accordance with the invention.

FIGS. 7 and 8 show a control table and a label table, respectively, for implementing feature access control in an illustrative embodiment of the invention.

10

Detailed Description of the Invention

The invention will be illustrated below in conjunction with an exemplary wireless communication system. Although particularly well-suited for use with, e.g., a telephone system which supports both wired deskset terminals and wireless terminals, the invention is not limited to use with any particular type of system or terminal. The disclosed techniques may be used in any communication application in which it is desirable to control terminal state in a processing-efficient and bandwidth-efficient manner. For example, the invention may be applied to handsets for use in cellular and personal communication services (PCS) systems, and to other types of communication terminals, such as wired ISDN terminals. The word "terminal" as used herein should therefore be understood to include not only portable wireless handsets as in the illustrative embodiment, but also other types of communication devices, including personal computers, wired and wireless desksets, optical communication terminals, or any terminal supported by a message-oriented command structure. It should be noted that the invention does not require any particular type of information transport medium, i.e., the invention may be implemented with any desired transport type. The term "switch" as used herein should be understood to include enterprise switches and other types of telecommunication switches, as well as other types of processor-based communication control devices such as servers, computers, adjuncts, etc. The term "table" as used herein is intended to include not only tabular representations as in the illustrative embodiments, but any other type and arrangement of data from which information can be extracted using one or more identifiers.

Information stored in an addressable memory is an example of one type of table that may be used in conjunction with the invention. The term "label" as used herein is intended to include text, graphics or other types of user-perceptible information, as well as various combinations thereof. A character string is an example of one type of label that may be used in conjunction with the invention.

FIG. 1 shows a portion of an exemplary communication system 100 in which the invention may be implemented. The system 100 includes an enterprise switch 110 which receives as an input a trunk 114. The trunk 114 supplies incoming calls to the switch 110 for processing. The switch 110 in this embodiment includes a central processing unit (CPU) 115, a memory 116, at least one interworking function (IWF) 117, and a system database 118. The CPU 115 may be a microprocessor, an application-specific integrated circuit (ASIC) or other type of digital data processor, as well as various portions or combination of such elements. The memory 116 may be a random access memory (RAM), a read-only memory (ROM) or combinations of these and other types of electronic memory devices.

The IWF 117 is used to provide necessary format conversions pertaining to signaling and transport, in a known manner. The IWF 117 may in other embodiments be incorporated into other elements of switch 110, such as the CPU 115 and memory 116. The system database 118 may be used to store, e.g., feature assignments to particular feature buttons, directory number assignments to corresponding call appearances or direct facility termination keys, access restrictions, and other known administrative information regarding the configuration of the system 100, as well as other types of information.

INS 22 *22* The switch 110 in this example further includes four port cards 120A, 120B, 120C and 120D. Port card 120A is coupled to a wireless base station 121 which communicates with a first wireless terminal (WT) 122 designated WT1 and a second wireless terminal 123 designated WT2. The terminal WT1 may be a mobile telephone, and the terminal WT2 may be a wireless deskset. Port card 120B is connected to a broadband wireless base station, e.g., a National Information Infrastructure (NII) wireless base station 124, which communicates with a wireless personal computer (WPC) 125. Port card 120C is connected to a wired deskset (DS) 126. Port card 120D is connected to an advanced terminal (AT) 127, which may be, e.g., a video telephone operating in

accordance with the H.320 standard. It should be noted that the switch 110 may include additional port cards, and may be connected to other types and arrangements of user terminals. The switch 110 is also connected to an administrator terminal 128 which may be used to program the operation of the switch 110 during a system administration, e.g., an initial set-up and configuration of the system or a subsequent system-level or user-level reconfiguration.

The system 100 of FIG. 1 includes an adjunct feature server 129. The adjunct feature server 129 may be directly connected to the switch 110 or connected thereto over a network or other suitable transport medium. The adjunct feature server 129 may be used, e.g., to implement state control logic for use in maintaining or otherwise processing a state transition table in accordance with the invention. Although shown as separate from the switch in the embodiment of FIG. 1, an adjunct such as adjunct feature server 129 is considered to fall within the general definition of the term "switch" as given previously. Such an adjunct may be physically incorporated within the switch in other embodiments of the invention, and may be partially or completely implemented using other switch elements such as CPU 115 and memory 116.

FIG. 2 shows a wireless terminal 122, which in the illustrative embodiment of the invention is configured to operate in accordance with a state-based control model. The terminal 122 includes a housing 150 with a speaker 152, a microphone 154, a display 156 and an audio alerter 158. The display 156, which may be an LCD display or other suitable type of display, includes a display area 160, a set of local icons 162-1, 162-2 and 162-3, a system icon 164, and a set of SLK labels 170-1, 170-2, 170-3 and 170-4 which indicate the functions associated with SLKs K1, K2, K3 and K4, respectively. A given physical SLK can have multiple function assignments which vary in accordance with the feature labels, based on the state-based control techniques of the invention to be described in greater detail below.

The local icons 162-1, 162-2 and 162-3 indicate locally-generated status information associated with the wireless terminal, e.g., battery charge remaining, signal strength, etc. The system icon 164 conveys system information supplied to the terminal by the switch. Alternative embodiments could include multiple switch-driven system icons. The wireless terminal 122 further includes buttons P1, P2, P3 and P4, LED indicators 172-1, 172-2 and 172-3, and a conventional set of touch-tone dialpad buttons 174. It should be emphasized that the configuration of wireless

terminal 122 as shown in FIG. 2 is for purposes of illustration only, and should not be construed as limiting the invention to any particular type of terminal.

The present invention in an illustrative embodiment provides techniques for controlling the SLKs and their associated labels on the wireless terminal 122. In accordance with the illustrative embodiment of the invention, at least a portion of a user interface state machine is downloaded into the wireless terminal 122 by the switch 110 at, e.g., terminal registration time, and during system operation relieves the switch of much of the overhead associated with operating the wireless terminal interface. The state machine may be in the form of a state transition table or other suitable representation of state transition information. The state machine of the illustrative embodiment resolves update contention by providing local updates based on state, relieves system congestion by maintaining SLK updates locally in the wireless terminal, resolves interpretive race conditions by maintaining an explicit set of state-based button/label interpretation assignments locally in the wireless terminal, and reduces bandwidth consumption by either reducing or eliminating system updates to the wireless terminal interface.

In the illustrative embodiment, the switch can, e.g., re-download the state machine if terminal operation is interrupted for any reason, and/or can update the state machine each time the terminal registers with the switch. The responsibility for maintenance of the state machine may alternatively be implemented entirely within the terminal, or using an approach in which the responsibility is shared between the switch and the terminal. In the latter case, either the switch or the terminal may be designated as a master.

When the wireless terminal 122 is administered, e.g., at system startup, the system 100 creates a state table based on user needs as expressed in feature requirements declared in a station administration form, and based on system knowledge of feature operations. For example, when a conference feature is administered on the SLKs of the wireless terminal 122, a consequential state is created by the system. This state represents those events that are legitimate for the user after activating a conference function, e.g., add another member to the conference, drop a member from the conference, transfer, etc. The transition from the original SLK label screen to the consequential feature state is defined, and the results loaded in the state table. Other features are treated in a

similar manner, resulting in the creation of a complete state table covering all of the desired features, and including access to local features.

During this process, feature identifiers that specify, e.g., the system code to access an associated feature, are inserted as data in the state table. These system-based feature identifiers are also referred to herein as system button identifiers (SBIDs). Note that SBIDs represent internal system identifiers for specific switch-based features. As noted previously, the table may also contain local feature identifiers. These local feature identifiers may be interleaved with the system feature identifiers. For example, as will be illustrated in conjunction with FIGS. 5, 6A and 6B, local features may be entered by way of an activity list specifying a subroutine, script, macro or other program to be executed if a user selects a particular SLK in a given state. This directs the terminal to go into a local mode, in which case some or all of its communication with the switch may be suspended. In the illustrative embodiment, when the system is activated and/or when the terminal registers with the switch, the state table is loaded into the wireless terminals. A copy of the table is generally also maintained in the serving switch.

At run time, when a given SLK is depressed, the wireless terminal sends the SBID associated with the currently-displayed label, and, assuming the state machine is locally controlled, updates its own display based on the next set of labels contained in its state table. In an embodiment in which the state machine is controlled by the switch, the wireless terminal may send to the switch an identifier of its current state as well as the SBID, and the switch returns an acknowledgment including an identifier of a new state. The wireless terminal then updates its display based on the next set of labels contained in its state table as specified by the new state identified in the acknowledgment.

FIGS. 3 and 4 are diagrams illustrating the operating modes and state transitions, respectively, for providing the above-described SLK control in the wireless terminal of FIG. 2 in accordance with the invention. FIG. 3 illustrates the four general operating modes of the wireless terminal 122 relating to the SLK control of the invention. These four operating modes include, e.g., a transitory standby mode 200 in which the wireless terminal is not engaged in any particular activity; a local mode 202 in which the wireless terminal is being manipulated by the user for the purpose of local programming or local feature service; a download mode 204 in which the switch

is downloading data into the wireless terminal; and a system mode 206, also referred to as a run-time mode, in which transport channel and feature service are available from the serving switch. In the download mode 204, communication is generally from the switch to the wireless terminal, with the uplink transport channel blocked but bidirectional signaling maintained for acknowledgments and the like, while in the system mode 206 full bidirectional communication between the switch and the wireless terminal is supported.

FIG. 4 shows a diagram in the form of a tree which illustrates state transitions in accordance with the invention. The nodes of the tree are in the form of blocks labeled L1A, L2A, L2B, etc. Each of these blocks corresponds to a particular set of SLK labels, in the general form of the labels 170-1, 170-2, 170-3 and 170-4 in FIG. 2. The first, second and third labels in each set are designated 1, 2 and 3, respectively. The fourth label in each set corresponds to a navigation function, and is designated with a "►" character. The notation used to identify the nodes in the tree includes a level identifier L_n , where n is the particular level, and a block identifier A, B, C, etc. A given state in the state-based SLK control of the invention corresponds to a node of the tree, and thus a particular set of SLK labels as well as a set of control data, e.g., SBIDs, related to that set of labels. Navigation within the tree is such that pressing an SLK label 1, 2 or 3 results in movement down the corresponding branch to the next node of the tree, while pressing the "►" character results in movement to the right, i.e., to another segment of the same branch of the tree or to another branch of the tree, depending on the implementation. The movements from state to state within the tree are arranged in accordance with a specified state transition table. Transitions from a current state to the next state may be expressed as follows:

$$\text{Current State} + \text{Transition (SBID)} = \text{Next State.}$$

Note that correlating the state with a particular set of labels and SBIDs eliminates the previously-described race condition problem.

FIG. 5 shows a portion of a state transition table which may be used to implement the above-described state-based SLK control in accordance with the invention. The table, which is only partially populated in this example, includes a column for the current state identifier (SID), a column

for SLK number, a column for SBID, a column for the new SID, and a column for an activity list. Associated with each of five possible current states in this example is a set of SLK labels, including a label for each of the four SLKs. For example, it can be seen from the FIG. 5 table that the SLKs 1, 2, 3 and 4 have been assigned labels 1A, 1B, 1C and 1D, respectively, in the state corresponding to SID 1. The table also includes an SBID for each of these assigned labels, i.e., SBIDs 11, 12, 13 and 14 for labels 1A, 1B, 1C and 1D. The new SID specifies the next state that the state-based control model moves to after the corresponding SLK is depressed by the user. The activity list may include additional relevant information for each entry, or may be eliminated in an alternative embodiment. As an example, the activity list may direct the terminal to enter a local mode and execute a preprogrammed local subroutine, script, macro or other program from a given state, after which the terminal returns to that state. The activity list may also direct the terminal to perform local functions such as memory updates, or attribute and condition adjustments.

FIGS. 6A and 6B show a more detailed example of a state transition table for controlling SLKs in a wireless terminal in accordance with the invention. This example assumes that the wireless terminal supports a total of seven primary states. These states are designated 1a, 2a, 3a, 4a, 5a, 6a and 7a. Primary states 1a, 2a, 3a, 4a and 5a each include two related states, designated Nb and Nc, where N is the primary state number. Each of the states provides a particular set of labels for four SLKs. For most of the states, three of the SLKs are used for functions or features, and one of the SLKs is used for navigation. The label indicating navigation in this example is the <more> label. The states and their corresponding sets of SLK labels in the example of FIGS. 6A and 6B are as follows:

1a. HOME State:

Dir Redial SAC <more> | 1b: f1 f2 f3 <more> | 1c: f4 f5 local <more> |

2a. Offhook State:

Conf Trans Hold <more> | 2b: f6 f7 f8 <more> | 2c: f9 f10 f11 <more> |

3a. Conference State:

Drop Add Hold <more> | 3b: f12 f13 f14 <more> | 3c: f15 f16 f17 <more> |

4a. Transfer State:

Trans Exit {blank} <more> | 4b: f18 f19 f20 <more> | 4c: f21 f22 f23 <more> |

5a. Hold State:

Conf Trans {blank} <more> | 5b: f24 f25 f26 <more> | 5c: f27 f28 f29 <more> |

6a. Directory State:

Next Cdisp Exit {blank} |

7a. Conference Max State:

Drop {blank} Hold <more> |

The set of SLK labels, SBIDs and transitions between the states are all specified in the state transition table of FIGS. 6A and 6B. The various system functions and features referred to in the state transition table, such as directory (Dir), send all calls (SAC), redial, conference (Conf), transfer (Trans), hold, call display (CDisp), etc. are well known in the art and will therefore not be described in detail herein. References to “hard buttons” for call appearances (CAs) refer generally to buttons such as P1, P2, P3 and P4 on wireless terminal 122 of FIG. 2.

The state transition table of FIGS. 6A and 6B allows the terminal to enter a local mode from state 1c, in which SLK No. 3, i.e., SLK K3 of FIG. 2, has a label “local.” When the user depresses this key, the terminal enters a local mode and, e.g., executes a preprogrammed local subroutine, script, macro or other program. As previously noted, the activity list portion of the state transition table may include an identifier of the local action to be taken, e.g., an identifier of the local program to be executed, although such an identifier is not explicitly shown in the table of FIGS. 6A and 6B. Alternatively, such an identifier could be included in another column of the table, such as the SBID column. The state transition table could include any desired number and arrangement of local and

switch feature labels. For example, a display line for a given state could include multiple local labels, intermixed with switch feature labels in any order. As another example, a given local program could be executed automatically upon entry into a particular state of the state transition table.

5 It should be noted that in the above-described illustrative embodiments, there are no interpretive race conditions, since system level feature identifiers, e.g., SBIDs, are transmitted to the switch explicitly based on current feature label contents. In addition, bandwidth consumption is greatly reduced, since the maximum response to given feature is a new state identifier, e.g., approximately one octet, compared to the dozens of octets of ASCII character strings typically associated with a display update in a conventional system. Furthermore, feature interpretation and display updates may be substantially eliminated for the system, since the feature identifiers sent by the wireless terminal to the switch do not need to be processed through a switch-based mapping function. Another advantage of the invention is that the SLK labels in the illustrative embodiment are fully language-independent, because there is a correspondence established between a character set in the terminal and the state transition table. This avoids the problems associated with preprogrammed labels, particularly language-related problems. It should be noted that, although illustrated using a wireless terminal, the invention can provide similar benefits in applications involving wired terminals.

Another aspect of the invention relates to providing feature access control using a bidirectional mapping function which allows single-occurrence features found on one terminal, e.g., a wired terminal such as terminal 126 of FIG. 1, to be represented multiple times on a different terminal which supports SLKs, e.g., a wireless terminal such as wireless terminal 122. This mapping function in accordance with the invention is implemented in an illustrative embodiment using two tables, a control table and a label table, which provide a set of interrelated pointers to execute the mapping function. The control table organizes groups of features into states, such as those described in conjunction with the state-based control aspects of the invention, and the label table stores label strings and associated attributes for each of the states.

FIGS. 7 and 8 show a control table and a label table, respectively, for implementing the above-noted feature access control in an illustrative embodiment of the invention. The control table

as shown in FIG. 7 includes a state identifier STATE_ID and a list of label identifiers (LIDs) which serve as pointers to the associated SLK labels for that state. The STATE_ID and the LIDs are each represented by a single octet, in this embodiment octet 4 for the STATE_ID and octet $n+4$ for LID n . It is assumed for this embodiment that octets 1, 2 and 3 are dedicated to protocol control information normally associated with message-based signaling. Although the control table of FIG. 7 shows the set of LIDs for a single state, the control table will generally include similar information for other states, e.g., a list of LIDS for each state in a state transition table such as that described previously.

Each of the LIDs in the control table of FIG. 7 points to a specific set of information in the label table of FIG. 8. This set of information in the label table may include, for a given LID, a character string corresponding to a given label, and a feature identifier, e.g., an SBID, corresponding to the internal system code for a given feature. It should be noted that FIG. 8 shows an illustrative download format for the label table, e.g., the label table information which is downloaded from the switch at terminal registration or system startup. The label table as actually stored in the terminal may include additional information not shown in FIG. 8, such as a presentation attribute for the current presentation state, e.g., on, off, blink, etc. The SBID is included in the label table in this embodiment because it can be used as an index into the label table in order to update stored presentation attributes.

The label table provides a single point of update for multiple occurrences of a single SLK label string. In the FIG. 8 example, each string is assumed to include four characters each, e.g., four ASCII-based international characters, although other types and arrangements of characters could also be used. Note that the label table of FIG. 8 shows the information for a single label string, i.e., the label string for LID n . This information includes LID n , represented by octet 4, an associated SBID, represented by octet 5, and four characters L_n_CHAR1 , L_n_CHAR2 , L_n_CHAR3 and L_n_CHAR4 , represented by octets 6, 7, 8 and 9, respectively. The label table may include similar information for all unique SLK label strings to be specified. For example, the SLK label string identified by LID 2 may be specified in octet 10, its corresponding SBID in octet 11, and its four characters in octets 12 through 15. The information for LIDs other than LID n is omitted from the FIG. 8 label table for simplicity and clarity of illustration. In the illustrative embodiment of the invention, the entire

control table and entire label table may be downloaded to the terminal from the switch, but other arrangements are also possible.

In operation, entire display lines of SLKs, e.g., a set of four labels 170-1 through 170-4 in terminal 122 of FIG. 2, may be written to the terminal 122 by sending a new STATE_ID to that terminal, along with a presentation attribute and the SBID of the last feature activated, if applicable. The terminal then interprets the set of LIDs specified for that state in the control table by referencing the label table and extracting the character strings and presentation attributes associated with that set of LIDs. The terminal can thereby accurately represent both the labels and the condition, as defined by the presentation attribute, of the current SLK display line. A situation in which the presentation attribute and SBID of the last feature activated need not be sent is if the activation is due, e.g., to a switch-side occurrence, rather than in response to a feature access indication from the terminal. In this case, a new STATE_ID may be sent without updating the presentation attribute of the last feature activated.

The above-described feature access control aspect of the invention provides a number of advantages over conventional systems. For example, the invention in the illustrative embodiment provides a fully-resolved bidirectional mapping, i.e., a one-to-many and many-to-one mapping, between single instances of switch features being represented as multiple instances of terminal features. The mapping supports a single point of update for all instances of a label string corresponding to a given switch feature which is accessed from multiple states in the terminal, and a single point of storage for all SLK label character strings, such that each string is stored only once. Entire SLK display lines can be updated by transmitting three short pointers, i.e., new STATE_ID, SBID of the most recently activated feature, and presentation attribute of that SBID, of one octet each instead of dozens of octets of displayable text as in a conventional system. This results in a considerable reduction in the bandwidth required to update SLK display lines.

The state-based control and feature access control in the illustrative embodiments of the invention may be implemented in whole or in part in a port card in the serving switch, e.g., in port card 120A associated with wireless base station 121 in system 100 of FIG. 1, elsewhere in the enterprise switch 110, e.g., using CPU 115 and memory 116, in the wireless terminal 122, in the adjunct feature server 129, or in various combinations of these and other system elements. Other

suitable arrangements of hardware and/or software may be used to implement the state-based control, and feature access control processes in accordance with the invention. The wireless terminal 122 includes suitable circuitry for receiving and decoding information received from the switch 110, and executing the corresponding commands. Such receiving, decoding and execution circuitry may include, e.g., a conventional processor and memory, and may be implemented in a straightforward manner, as will be apparent to those skilled in the art.

An automated administration system for generating a state machine in accordance with the invention will now be described. As noted above, the state transition table or a suitable representation thereof may be loaded by the switch into the terminal at terminal registration time, system startup, system administration time, etc. Although the system administrator may manually create the necessary relationships between the various system features and the state machine that is loaded into the terminal, in many applications it will generally be more reliable and efficient to utilize an automated administration system in accordance with the invention. When creating user interface state machines on a per-user or per-group basis, a human operator, responsible for up to several thousand terminals, may not be capable of following a complex set of written rules to ensure reliable creation of the state machines. During the definition of the state transition tables, this set of rules about which features have antecedent or descendant relationships is applied to ensure proper transition operation, and after operational integrity is assured, additional rules pertaining to consistency of presentation are enforced. The antecedent or descendant relationships may be characterized in the form of a tree such as that shown in FIG. 4, and examples of such relationships can be seen in the state transition table of FIGS. 6A and 6B.

The automated administration aspects of the invention provide a mechanism for automating the creation of the state machine for controlling terminal SLKs. This mechanism will be illustrated for creation of a state transition table which fully characterizes the state machine, but it should be recognized that the invention is similarly applicable to the creation of other representations of state machine information. The automated administration process for the state transition table is as follows:

1. The user or a representative provides a completed terminal subscription form or other similar representation indicating desired features, layout and language. The terminal subscription

form may be completed by entering information via a system terminal, system-attached terminal or other suitable information-entry mechanism.

2. A system administrator enters the data from the terminal subscription form into the administration system.

3. As the data is entered, the automated administration system of the invention performs the following operations:

a. Checks a system database, e.g., database 118 or other system storage element, to establish feature type, e.g., simple/reflexive or complex/descendant, and extracts a system feature identifier, e.g., SBID, and character string for the corresponding label.

b. Checks the label table to see if there is an entry already present for the extracted SBID. If no entry is found in the label table, the system assigns a LID to the feature, and inserts that LID into the list of LIDs for the current STATE_ID in the control table. The current STATE_ID may be derived, e.g., from the position of the feature on the subscription form. The system then updates the label table with the new LID, SBID, and a character string extracted from the system database based on the specified language. If an entry is found in the label table for the SBID, the system extracts the LID from the label table and assigns it to the next open position for the current STATE_ID in the control table.

c. If the feature type is complex/descendant, the system checks the database for the descendant relationship definition, and creates a new STATE_ID for that descendant.

d. Repeats operations a, b and c until all entries from the terminal subscription form are entered, and all of the descendants have a corresponding STATE_ID.

e. When all entries from the terminal subscription form have been processed, the system executes an operation, e.g., a sort operation, to ensure positional consistency or other desirable human factors arrangement among LIDs across all states and to compress blank areas where possible without violating the positional consistency criteria.

The automated administration process described above is generally repeated for each individual user or for groups of users, based on the information in the terminal subscription form for a given user or group. As a result, each user or group of users can be provided with a unique state machine and thus a unique arrangement of features accessible via their SLKs. The operations given

